May 1991 ... IBM TECHNICAL DISCLOSURE BULLETIN ... <-- pp113-120 -->

Title:

Technique for Replicating Distributed Directory Information.

Text

TDB

A technique is disclosed which facilitates the subsetting and replication of directory information in a distributed environment. The technique, which is not dependent on the network topology or data content, provides subsetting of the directory information. Through replication, it facilitates the management and control of the data while providing flexibility for satisfying differing requirements and balancing the trade-off between the amount of data storage and performance required.

Also disclosed is the application of the technique by the directory service to allow it to recursively manage its own information in the same manner that it manages other directory information.

The objects contained within a computer based directory are frequently objects known or managed by the computers in the network, or by the users of those computers. Examples of the former might include objects such as the computers themselves (sometimes called "nodes"), the users themselves (sometimes denoted by "user identification"), groups of data or information, such as computer files or data bases, or the information in the directory itself. Examples of the latter also include objects known to the user of the computer system, but not necessarily known or managed by the computer, such as the postal or residence address of individuals.

With the diversity of objects to be included within a computerized directory, it is apparent that the characteristics of the objects may differ greatly and that the requirements for each object may even conflict. For example, the residence location of an individual is normally stable and changes infrequently, while the existence and location of a computer data file may be extremely volatile and transitory. The requirements for dissemination of updated information and the ability to be tolerant of temporary inaccuracies of the data differ for users of the two different types of directory data.

A computer-based directory in a distributed environment must be able to accommodate a broad diversity of objects about which it contains information, and to facilitate the directory's ability to satisfy greatly differing or conflicting requirements pertaining to the objects.

In order to describe the disclosed technique for subsetting and replication of directory information, it is convenient to first define several terms.

The collection of all information contained in the directory can be organized as disjoint named subsets, called Partitions. A Partition

Name, which identifies a collection of directory entries that can be independently accessed, distributed or administered, is structured such that

```
Partition Name = Class. Level.
Partition_Name_Qualifier(s)
where:
Class distinguishes between the defined types of
directory service classes, e.g., electronic mail, telephone,
data file, data base or directory objects.
```

Level allows a structure to be defined for a specific directory Class and supported by the directory service, e.g., Public, Shared and Private Levels of an electronic mail directory service class.

Partition Name Qualifier(s) are tokens which assure network-wide uniqueness of the Partition Name. For certain Classes the values would normally be supplied by the user creating the partition. For other Classes, the values are generated by the directory service.

Each partition also may have one or more Partition Name Aliases. This alias is a "user friendly" name that may be applied to a partition (or a group of partitions) when it is to be referenced by users.

The collection of information contained in a partition is a set of one or more Partition Entries, chosen to facilitate the management and control of the data. The Partition Entry, which is the collection of information about a specific instance of an object contained in the directory, also has a unique Partition Entry Name which is structured as

```
Partition Entry Name = Resource . Row,
where
Resource identifies the resource type within the
Class.
Row is the unique identifier for the individual entry.  For
certain Resource types, this value would normally be supplied b
user creating the entry.  For other Resource types, the values
generated by the directory service.
```

Each Partition Entry has a Network Unique Identifier, which is composed of the Partition Name together with the Partition Entry Name.

Network Unique Identifier = Partition Name . Partition Entry Name

The information within an occurrence of a partition entry may include one or more network unique identifiers for related entries. The related entries are allowed to be in other (either locally or remotely located) partitions.

Each partition entry may also have one or more Partition Entry Aliases. This alias is a "user friendly" name that may be applied to a partition entry (or a group of entries) when it is to be referenced by users.

To describe replication of partitions, it is convenient to also define the concept of "master" and "shadow." The Master Partition Entry contains the original directory information for the entry. There can be only one Master Partition Entry for a specific entry. A Master Partition then is a directory partition in which each of the entries is a Master Partition Entry. It follows that there can be only one "master"

of a partition. A Shadow Partition Entry is a copy of a Master Partition Entry. Similarly, a Shadow Partition is a partition containing shadow partition entries for each of the partition entries in the corresponding master partition. Throughout a distributed network, a master partition may have zero, one or multiple shadow partitions (or copies of itself).

Authors of computer-based directories in a distributed environment are confronted with many diverse, often conflicting, requirements for managing the information about objects within the directory. The technique disclosed uses a naming algorithm, as defined above, to allow organizing the set of all objects contained in the directory into disjoint subsets, or partitions. It facilitates the subsetting into partitions based on a number of criteria, including:

Class whereby types or categories (e.g., electronic mail, directory objects) of directory data having consistent requirem (e.g., for response time characteristics, or tolerance for temporarily back level data) can be addressed.
Level whereby differing requirements for a specific directory class must be addressed (e.g., an electronic mail dir satisfying requirements for a Public, Shared (or Workgroup) and Private directories).

Further, it allows the directory service or the administrators of the directory to create subsets (partitions) in response to other requirements for managing and controlling the data, such as the following examples:

Response time requirements necessitating the availability of the partition locally,
Data storage constraint requirements forcing a portion of the information (some partitions) to be remotely located,
Geographic or organizational constraints on the location of a partition,
Administrative or security requirements affecting the grouping of partition entries into various partitions,
The requirement to implicitly limit the scope of a query against the directory to those entries within selected partitio

The naming technique disclosed facilitates satisfying requirements such as these in whatever is the most effective manner for the customer's requirements and network configuration without placing undue constraints on the customer. While the definitions of Partition Name and Partition Entry Name apply to a selected customer's network, they are readily extensible across geographic and organizational boundaries to satisfy worldwide interconnected networks.

As an example, a directory service could be produced to support the requirements of an electronic mail system within a company. For this situation (Class = Electronic Mail), there might be differing requirements that are implemented using three hierarchical levels:

At the highest level (Public, for example), the directory partitions might contain partition entries for each individual the domain of the electronic mail system. They might also cont partition entries for departments within the company, where eac entry includes the network unique identifier of each member of

department, hence establishing relationships among the partitio entries.
At the middle level (Shared, for example), the directory partitions might contain partition entries for members of vario workgroups. In this case, the partitions might contain partiti entries for members of the workgroup (which would probably cont the network unique identifier for the individual within the Pub level). They might also contain entries for other individuals (customers, for example) not contained within the Public (or co level.
At the lowest level (Private), each partition might belong to an individual in the company. In this case, it is likely that partition entries would be primarily "aliases" (or "nicknames") individuals or groups of individuals (sometimes referred to as distribution lists) with whom the owner of the Private partitio frequently communicates. In this case also, most of the entrie might contain network unique identifiers for the desired entrie either the Public or Shared level partitions.

This example illustrates the manner in which differing requirements can be supported using the described technique. It also illustrates the use of relationships (i.e., the inclusion of the network unique identifier in a partition entry) to "point to" the data, rather than requiring the duplication of the data at multiple locations, with the inherent problems of distributed maintenance as the data is changed.

Fig. 1 illustrates the logical subsetting of directory information into partitions.

If the directory were limited to a single computer node, all the partitions could simply be master partitions, with all being locally resident on the computer. If, on the other hand, the directory is distributed over several computers, there are additional requirements for availability of the data which arise. The disclosed technique, again using the naming algorithm defined above, allows the creation and placement of shadow partitions. In this case, the "Partition_ Name_Qualifiers" must indicate whether the partition is a master or shadow partition. With this technique, the customer can place his directory partitions throughout the network in a manner which best satisfies his needs. The placement capabilities are:

Fully distributed (that is, only master partitions exist in the network).
Partially replicated (that is, shadow partitions of some master partitions exist in some, but not all, nodes in the network).
Fully replicated (that is, shadow partitions of all master partitions exist so that every node in the network contains a complete copy of the total information).

This flexibility in placement capabilities allows the customer to make his own trade-off with regard to storage and response time. Fig. 2 illustrates the three degrees of replication in a simple network.

The techniques disclosed above are particularly powerful when applied to a directory class used by the directory service itself. The application of the technique for defining a partition, where the partition entries are the information defining the type (master and shadow) and placement (where located) of all directory partitions, is disclosed. For simplicity, refer to such a partition as a Directory of

Partitions (DOP).

With this definition for a Directory of Partitions, the algorithm illustrated in Fig. 3 can readily be used at a given node to determine the location of the "nearest" partition containing the desired data. A query from a requestor is normally preprocessed and then presented to the directory service. Upon receipt of the query, the directory service first determines whether the partition (XYZ in the figure) containing the needed data is available locally (at this node). If so, the data is accessed and the query response is constructed and returned to the requestor. If the needed partition is not available locally, the directory service uses the information from the Directory of Partitions to determine the "nearest" location at which the data resides. The determination of "nearest" may take into consideration a number of factors, such as distance, bandwidth of the telecommunications links, tariff structures, or availability of alternative remote computer systems. Having determined a location at which the needed data resides, the directory service forwards the query request to the selected remote location. Upon receipt at the remote location, the same algorithm can be applied to satisfy the query.

Thus, the definition of directory partitions and replication of those partitions can be done in such a manner that it is not dependent on the topology of the network or the content of the data. With an algorithm to use the information from the Directory of Partitions, the node at which requested directory information is located can readily be determined.

Diagrams:

FIG. 1

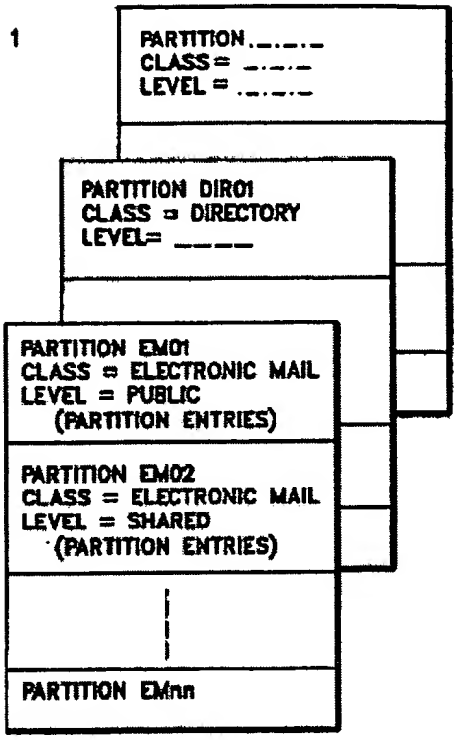
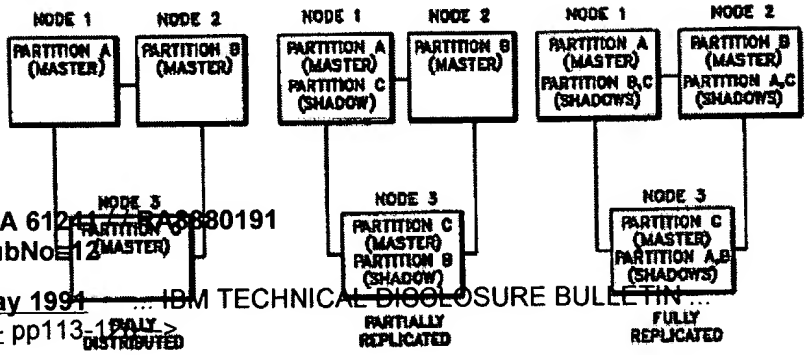
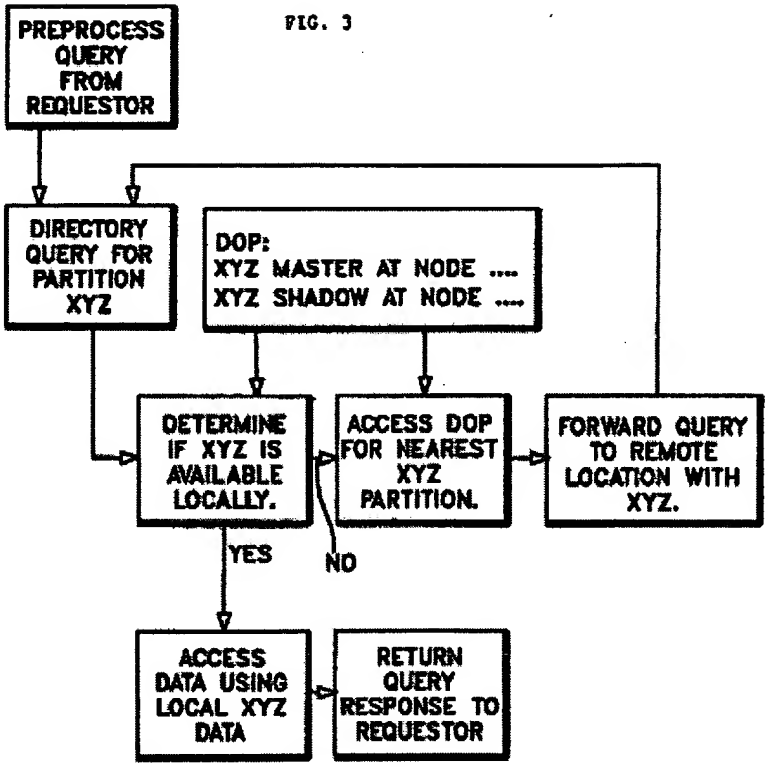


FIG. 2



Order/Fcode/Docket: 91A 61241 / FA3880191
PubNo=12
: May 1991 ... IBM TECHNICAL DISCLOSURE BULLETIN ...
<-- pp113-117 ->

FIG. 3


PREPROCESS QUERY FROM REQUESTOR
DIRECTORY QUERY FOR PARTITION XYZ
DOP:
XYZ MASTER AT NODE ....
XYZ SHADOW AT NODE ....
DETERMINE IF XYZ IS AVAILABLE LOCALLY.
ACCESS DOP FOR NEAREST XYZ PARTITION.
FORWARD QUERY TO REMOTE LOCATION WITH XYZ.
YES
NO
ACCESS DATA USING LOCAL XYZ DATA
RETURN QUERY RESPONSE TO REQUESTOR